Critical Analysis

The positive aspects of my submission is that I found it easy to remember what data structure is required for this to be able to function and sort, so I took the basic LinkedListGen format from assessed exercise 1\_2 Task B to be able to use it here in this format along with all the functions like adding and insertinorder.

The negative aspects was figuring out how to do the Available function as I was thinking I should be using an altered version of the selectedsort and I’m unsure if you can do it like that. But I decided to go a different route to find the solution by making my own function to display the optimal amount of things to attend with the information entered into the GUI.

Some key difficulties I had was being unsure on how to tackle this task, I asked matteo.c on

Discord where to start and was led to the second application of the Greedy Algorithms video to watch it again and this helped me get a grasp on how to start the task.

My methodology on how the Available function works is to first initialize a linkedlist in the function using the LinkGen with the type of Request to firstly create a linkedlist and with the type of request to be able to use the methods in the request class which involves the things we want to be able to access like StartTime, ID and EndTime which is what we are sorting by so we can say if the starttime Is greater than the endtime, don’t add it. I am also overriding the ToString() function so that it will print out the ID, StartTime and EndTime and also change the format of it to make it more presentable and organized, then in the Form1.cs we are then iterating through the linkedlist and check if the starttime of the current section of the linkedlist is on with the endtime and make sure that the starttime is larger than the endtime. If this is true print out the temp.Data to a string and making the string endtime equal to the temp.Data.EndTime and then display the buffer and call the Available function through the display button.

There are some things that can be improved in my submission by removing the redundant code that is not used such as inorder, postorder and more to reduce the un necessary length of the code to make it simpler to understand.